Amendments to the Specification:

The following amendments are made to the specification to overcome the objection to informalities in the drawings.

The following amendments to the specification are made on pages 7-8, paragraphs 19-20 to overcome the objection to Fig. 2, reference numerals 208 and 240. Please replace paragraphs 19-20 on pages 7-8 with the following paragraphs.

[0019] Fig. 2 shows an exemplary diagram of an intermediate software layer 220 positioned between a user interface 210 and a layer of business logic 230. The intermediate layer 210 can collect larger amounts of the data from the UI that may be typically collected prior to processing by the BL. Hence, the data does not have to be entered serially in the UI. The larger amounts of data can be formatted in the intermediate layer 210 and sent to the BL layer 230. As a result, the data does not have to be serially processed and buffered in the BL layer, but rather can be handled in "batch" mode, that is, all at once without intervening human interaction. In one case, the intermediate layer 220 can provide temporary buffering of data until the user has completed entering information for a "batch" of business processes. As a result, the intermediate layer 220 can improve the efficiency of the business system and can allow the UI 210 to be formatted so that a user can enter larger amounts of data, which can be processed in the BL layer and sent to the database 240.

[0020] The intermediate layer 210 may support one or more operations on multiple interdependent attributes of complex business objects. For instance, the UI 210 may have data relating to multiple processes 202, 204, 206, 208 all of which may interact with the intermediate layer 220.

The following amendments to the specification are made on page 11, paragraphs 28-29 to overcome the objection to Fig. 4, reference numerals 414, 416, 420, 424, 426, and 432. Please replace paragraphs 28-29 on page 11 with the following paragraphs.

7

[0028] In the example shown in Fig. 4, a user can initiate a query 410 in the UI 402, and the MAC 404 can send a request to the UI 402 for data. The data flow in Fig. 4 is initiated or triggered by an action in the user interface. The user enters the data in the UI 402. The UI 402 sends the data 412 to the MAC 404, associates that data in an object class, and sends the data 413 to the intermediate layer 406. The intermediate layer 406 translates the data and sends the data to the BL 408 at 414. The BL 408 may then request additional data from the UI by sending a request through the intermediate layer 406 to read data at 416 to send to ~~and~~ the MAC 404 (object model controller).

[0029] The MAC 404 may also have the functionality to "lock" the object 418. The "lock" function may prohibit other users of a system in other user interfaces from modifying the object. The user may select one specific object (e.g., an employee number object), and specify in the UI 402 that object modifications are to be locked from other users. In this example, after the processed data is sent to the BL at 420, locked objects may only be modified 422 by the user of the interface. When the user elects to save the data (e.g., by pressing a "save" button in the UI 402) after the modified data has been sent to the IL at 424 and BL at 426, the save request 428 is sent to the MAC 404. The MAC 404 can notify 430 the intermediate layer 406 that all data changes are to be saved. All data can be sent to the BL at 432. The BL 408 sends the data to be saved in a database 434. If a user sets an object lock 418, then the object can be unlocked 430 when or after the data is saved.

The following amendments to the specification are made on page 13, paragraphs 34-35 to overcome the objection to Fig. 5A, reference numerals 542, 560, 564, 566, 568, 559, and 569. Please replace paragraphs 34-35 on page 13 with the following paragraphs.

[0034] In the event that a user modifies data at 542 (e.g., employee number "54321") for the same object (e.g., object "A"), the object's state is changed 548 (e.g., A1 → A2). Because operations are performed on the same object, the intermediate layer will not change the focus 550. The intermediate layer 554 can instruct the BL to discard the previous state changes to the object "A" 554 and to start a new trial 556. The intermediate layer 554 can then send the modified state 558 (e.g., A0 → A2) to the BL.

[0035] In the event that a new object is used (e.g., object "B") from an entry into the UI at 560, the intermediate layer's focus can be changed 562, 563, and the same steps described above (e.g., steps shown in 534-539) can be performed on the new object at 564-569. The new object may represent, for example, an employee's start date or organizational title/ranking. Hence, the intermediate layer 554 may be able to recognize different objects (e.g., objects "A" and "B") and produce changes on those objects (e.g., A0 → A2).

The following amendment to the specification is made on pages 13-14, paragraph 36 to overcome the objection to Fig. 5B, reference numeral 578. Please replace paragraph 36 on pages 13-14 with the following paragraph.

[0036] Fig. 5B shows the intermediate layer operations when a user has completed the steps of entering data in the UI. The user may "save" all of the data and data modifications (e.g., by selecting a "save" button in the UI). The MAC 510 (object model controller) can send a "save" or "flush" request to the intermediate layer 570. The intermediate layer 512 can instruct the BL 512 to approve previous user entries and/or changes 574, and to flush or send the data and/or objects to a database 576. The intermediate layer 512 may also be responsible for initializing (or re-initializing) the system/framework at 578 so that a user can be allowed to start the data entry steps over again.